SWEBOK KA #10: Software Quality

The Software Engineering Body of Knowledge (SWEBOK) features 11 knowledge areas (KAs). The tenth KA is Software Quality. The Software Quality KA is focused on software quality considerations that transcend the life cycle processes. It includes three topics, as shown in Figure 1. These topics are Software Quality Fundamentals, Software Quality Management Processes, and Practical Considerations.

The Software Quality Fundamentals topic covers the underlying meanings of quality concepts and characteristics. The software engineering culture must be rooted in ethics. There is a cost and value of quality. The cost of quality is composed of prevention cost, appraisal cost, internal failure cost, and external failure cost. Models of quality characteristics can be useful for discussing, planning, and rating the quality of software products. Quality improvement occurs through an iterative process which requires management control, coordination, and feedback from many concurrent processes.

The Software Quality Management Processes topic discusses processes that help ensure better software quality. Software quality assurance processes provide assurance that the software products and processes in the project life cycle conform to their specified requirements. Verification and validation processes address software product quality directly and use testing techniques which can locate defects. Reviews and audits are broken down into five types: management reviews, technical reviews, inspections, walk-throughs, and audits.

The Practical Considerations topic addresses the practical aspects of software quality. Software quality requirements involve many aspects, including influence factors, dependability, and integrity levels of software. Defect characterization leads to an understanding of the product, facilitates corrections to the process or the product, and informs project management or the customer of the status of the process or product. Software quality management techniques can be categorized as static, people-intensive, analytical, dynamic, and testing. Static techniques involve examination of the project documentation and software without executing them. People-intensive techniques include reviews and audits. Analytical techniques include complexity analysis, control flow analysis, and algorithmic analysis. Dynamic techniques are testing techniques that include modeling and simulation. Finally, software quality measurement determines the degree of each quality characteristic attained by the product.



Figure 1. Breakdown of Topics for the Software Quality Knowledge Area